Docket No. QN1067.US
EXPRESS MAIL NO. EV305038709US

METHOD AND SYSTEM FOR PROVIDING DATA INTEGRITY IN STORAGE SYSTEMS

INVENTOR(S):

DHARMA R. KONDA

KATHY K. CABALLERO

SANJAYA ANAND

ASHISH BHARGAVA

RAJENDRA R. GANDHI

Cross Reference to Related Applications:

[0001] This application claims priority claim under 35 U.S.C.§ 119(e)(1) to the provisional patent application filed on August 27, 2003,Serial number 60/498,384, entitled "**METHOD AND SYSTEM FOR PROVIDING DATA INTEGRITY IN STORAGE SYSTEMS**", the disclosure of which is incorporated herein by reference in its entirety.

BACKGROUND

1. Field of the Invention

[0002] The present invention relates to storage systems, and more particularly, to maintaining data integrity in storage systems.

2. Background of the Invention

[0003] Conventional storage systems (disk drive etc.) store data bytes in sets of predetermined length. Disk

array storage systems have multiple storage disk drive devices that are arranged and managed as a single mass storage system. Redundancy is often used to aid data availability, where data or data relationship is stored in multiple locations. In the event of a failure, redundant data is retrieved from the operable portion of a system and used to regenerate the lost data. A RAID (Redundant Array of Independent Disks) storage system is one such system that uses a part of the physical storage capacity to store redundant data.

[0004] Data is typically moved from plural host systems (that include computer systems, and embedded devices etc.) to the storage system through a RAID controller.

[0005] Various standard interfaces are used to move data from host systems to storage devices. Fibre channel is one such standard. Fibre channel (incorporated herein by reference in its entirety) is an American National Standard Institute (ANSI) set of standards which provides a serial transmission protocol for storage and network protocols such as HIPPI, SCSI, IP, ATM and others. Fibre channel provides an input/output interface to meet the requirements of both channel and network users.

[0006] Host systems often communicate via a host bus adapter ("HBA") using the "PCI" bus interface. PCI stands for Peripheral Component Interconnect, a local bus standard that was developed by Intel Corporation ®. The PCI standard is incorporated herein by reference in its entirety. Most modern computing systems include a PCI bus in addition to a more general expansion bus (e.g. the ISA bus). PCI is a 64-bit bus and can run at clock speeds of 33 or 66 MHz.

[0007] PCI-X is a standard bus that is compatible with existing PCI cards using the PCI bus. PCI-X improves the data transfer rate of PCI from 132 MBps to as much as 1 GBps. The PCI-X standard was developed by IBM®, Hewlett Packard Corporation ® and Compaq Corporation® to increase performance of high bandwidth devices, such as Gigabit Ethernet standard and Fibre Channel Standard, and processors that are part of a cluster.

[0008] The iSCSI standard (incorporated herein by reference in its entirety) is based on Small Computer Systems Interface ("SCSI"), which enables host computer systems to perform block data input/output ("I/O") operations with a variety of peripheral devices including disk and tape devices, optical storage devices, as well as printers and scanners. A traditional SCSI connection between a host system and

peripheral device is through parallel cabling and is limited by distance and device support constraints. For storage applications, iSCSI was developed to take advantage of network architectures based on Fibre Channel and Gigabit Ethernet standards. iSCSI leverages the SCSI protocol over established networked infrastructures and defines the means for enabling block storage applications over TCP/IP networks. iSCSI defines mapping of the SCSI protocol with TCP/IP.

[0009] The iSCSI architecture is based on a client/server model. Typically, the client is a host system such as a file server that issues a read or write command. The server may be a disk array that responds to the client request.

[0010] When data is moved to/from host systems to/from disk storage systems at high data rates, (e.g., 2GBps), it is essential to maintain data integrity to take advantage of the high bandwidth that is offered by current industry standards.

[0011] Cyclic redundancy check ("CRC") is one way to maintain and validate data integrity. CRC bytes are generated and stored for each data set. CRC involves a process that operates on a block of data and generates a number (called checksum) that represents the content and organization of the data block. CRC is performed

on data so that by comparing the checksum of a block of data to the checksum of another block of data, an exact match can be found. CRC is performed when data files are transferred from one location to another (host to storage/storage to host).

[0012] CRC calculations themselves are well known in the art. However, conventional techniques do not provide complete data integrity via CRC because often CRC is performed either too late or too early in the data transfer process.

[0013] Therefore, there is a need for a system and method that can provide data integrity for modern storage systems that are operating in high band-with environment.

SUMMARY OF THE INVENTION

[0014] In one aspect of the present invention, a method for performing data integrity process is provided. The process includes selecting a cyclic redundancy code ("CRC") mode from amongst append, validate and keep, and validate and remove mode.

[0015] If the append mode is selected, then CRC is appended after each data block boundary. A CRC seed value is incremented for each data block providing a unique CRC value for each data block.

[0016] If validate and keep mode is selected, then CRC accompanying any data is compared to CRC that may have been accumalated and if an error occurs after the comparison, an interrupt is generated.

[0017] If validate and remove mode is selected, then CRC is first validated and then CRC is removed before data is sent out.

[0018] In yet another aspect, a system for performing data integrity process is provided. The system includes

[0019] CRC logic that allows firmware running on an adapter to select one of plural CRC modes including append, validate and keep, and validate and remove mode.

[0020] During append mode, a CRC engine determines the CRC for each data block and CRC seed value is incremented for each data block such that each data block has a unique CRC value. Also, each data block has a CRC value and an optional field where custom information may be added ("info data").

[0021] During the validate and keep mode, the CRC engine compares CRC for the data with accumalated CRC information and CRC is sent out with data.

[0022] During the validate and remove mode, the CRC engine compares CRC for the data with accumalated CRC

information and CRC information is removed before data is sent out.

[0023] In yet anorther aspect, an adapter in a RAID controller that is coupled to a host on one side and a storage media on another side is provided. The adapter includes CRC logic that can perform data intergrity process using one of plural CRC modes including append, validate and keep, and validate and remove mode. The CRC logic is functionally coupled to a PCI and/or PCI-X interface.

[0024] In one aspect of the present invention, data integrity is maintained through out the data path.

[0025] This brief summary has been provided so that the nature of the invention may be understood quickly. A more complete understanding of the invention can be obtained by reference to the following detailed description of the preferred embodiments thereof concerning the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0026] The foregoing features and other features of the present invention will now be described with reference to the drawings of a preferred embodiment. In the drawings, the same components have the same reference numerals. The illustrated embodiment is

intended to illustrate, but not to limit the invention. The drawings include the following Figures:

[0027] Figure 1 shows a system with an adapter, according to one aspect of the present invention;

[0028] Figure 2A shows an adapter as used in a RAID controller, according tone aspect of the present invention;

[0029] Figure 2B shows a format used in the CRC process, according to one aspect of the present invention;

[0030] Figure 3 is a block diagram of PCI interface components, according to one aspect of the present invention;

[0031] Figure 4 shows a block diagram of a system performing data integrity checks in the receive path, according to ne aspect of the present invention;

[0032] Figure 5 is a flow diagram of a system performing data integrity checks in the transmit path, according to one aspect of the present invention;

[0033] Figure 6 is a flow diagram of executable process steps for performing data integrity tests, according to one aspect of the present invention; and

[0034] Figures 7A-7B show various register values that are used to perform data integrity tests, according to ne aspect of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0035] To facilitate an understanding of the preferred embodiment, the general architecture and operation of a system using storage devices will be described. The specific architecture and operation of the preferred embodiment will then be described with reference to the general architecture.

[0036] It is noteworthy that a host system, as referred to herein, may include a computer, server or other similar devices, which may be coupled to storage systems. Host system includes a host processor, memory, random access memory ("RAM"), and read only memory ("ROM"), and other components.

[0037] Figure 1 shows a system 100 that uses a controller/adapter 106 (referred to as "adapter 106) for communication between a host system (not shown) with host memory 101 to various storage systems (for example, storage subsystem 116 and 121, tape library 118 and 120) using fibre channel storage area networks 114 and 116.

[0038] Host system communicates with adapter 106 via a PCI bus 105 through a PCI interface 107. Adapter 106 includes processors 112 and 109 for the receive and transmit side, respectively. Processor 109 and 112 may be a RISC processor.

[0039] Transmit path in this context means data coming from host memory 101 to the storage systems via adapter 106. Receive path means data coming from storage subsystem via adapter 106. It is noteworthy, that only one processor can be used for receive and transmit paths, and the present invention is not limited to any particular number/type of processors.

[0040] Processors 109/112 also include receive side/transmit side sequencers (referred to "SEQ").

[0041] Adapter 106 also includes fibre channel interface (also referred to as fibre channel protocol manager "FPM") 122 and 113 in receive and transmit paths, respectively. FPM 122 and 113 allow data to move to/from storage systems 116, 118, 120 and 121.

[0042] Figure 2A shows a block diagram of a system using RAID controller 200 coupled to HBA 204 via adapter 201 and storage system 203 via adapter 202.Data 205 is sent to storage 203 via adapters 201 and 202.

[0043] The receive path is shown as 208 and 207, where data moves from storage 203 via adapters 202 and 201 to HBA 204, and the transmit path is shown as 205 and 206.

[0044] In one aspect of the present invention, data integrity is maintained through out the data path.

[0045] Figure 2B shows a block diagram of data that is moved using system 210. Data 211 is followed by CRC

bytes 212. An optional information field ("Info") data 213 is also provided, which allows a user of system 210 to include custom information. Data block 211 may be 512 bytes, CRC 212 is 4 bytes and info data 213 can be another 4 bytes. It is noteworthy that the various adaptive aspects of the present invention are not limited to any particular block size.

[0046] Figure 3 is a block diagram showing PCI interface 107 components, as used in the adaptive aspects of the present invention. PCI interface 107 includes direct memory access ("DMA") and arbitration logic and is operationally coupled to PCI bus 105 at one end and to fibre channel wire 311 at the other end. Frame Buffer ("FB") 308 is used store information, when data moves from a host to storage system and vice-versa.

[0047] PCI interface ("PCI I/F) 107 also includes CRC logic 307 that performs various operations, according to the adaptive aspects of the present invention, described below.

[0048] PCI I/F 107 is also coupled to a receive path DMA unit (RDMA) 306 and a transmit side DMA (TDMA) 305 that provides DMA access to move information back and forth in the transmit and receive paths.

[0049] PCI I/F 107 is also coupled to various other DMA units, for example, command DMA unit 303, request DMA unit 303 and response DMA unit 302. These DMA units allow the use of PCI I/F 107 to move information in and out of adapter 106 by using standard DMA techniques.

[0050] RDMA 306 and TDMA 305 modules use various register values to execute the adaptive aspects of the present invention, as described below and also shown in Figures 7A and 7B.

[0051] Figure 4 shows a block diagram of system 400 that is incorporated in CRC logic 307 for the receive path. Data enters system 400 through FB 308 and is placed in a receive FIFO register (or storage) 412. Data is then sent to a multiplexor 413 (via receive pipeline register 411, if timing synchronization is needed).

[0052] Data is aligned by alignment logic 416, before being sent out to PCI bus 105.

[0053] In one aspect of the present invention, various modes may be used to implement data security techniques. The firmware running on processor 112 can select the CRC mode. In one aspect of the present invention, CRC may be implemented using an "append", "validate & keep" and "validate & remove" mode. A user

of adapter 106 can use a particular mode depending on how adapter 106 is being used. The following describes the various modes, according to one aspect of the present invention:

[0054] "Append" Mode: In this mode, CRC is appended to data 308 before being sent out to PCI bus 105. In this case, CRC is calculated by CRC engine 401 after each data block. Processor 112 provides CRC seed. Every block of data has CRC seed value. In one aspect of the present invention, CRC seed value is incremented, using counter 405 and that provides a unique CRC for each data block. CRC seed value is provided by processor 112 for the receive path. Processor 112 also provides accumulated ("ACC") CRC values 408 for data stored in a storage system.

[0055] CRC engine 401 generates the CRC 410, which is sent to CRC ACC register 403. Accumulated CRC values are sent to processor 112 and CRC engine 401. CRC error 409, if any, is sent to processor 112, while CRC 402B is sent out via residue register 414 and PCI bus 105.

[0056] In one aspect of the present invention, CRC seed 407 is incremented for each block of data ("Increment Mode", see Figures 7A and 7B). This allows the system to have a unique CRC for data block. In the

increment mode, CRC 212 and info data 213 follow the data together. CRC seed increments after each block of data.

[0057] "Validate and Keep Mode": During this mode, data 417 is sent via PCI bus 105 and CRC 418A is sent to CRC engine 401. CRC engine 401 compares CRC 418A with the CRC that it has accumulated in register 403, in real time while data 417 is moving out to PCI bus 105. If there is an error, an interrupt is sent to processor 109/112. During this mode, CRC from the storage system is also sent to PCI bus 105.

[0058] "Validate and Remove Mode": In this case, data comes from FB 308 and is validated, as described above. After the validation, CRC is removed and only data is sent to the host. Hence, the host and the storage systems are not involved in the data integrity process. If any errors occur then they are reported to RISC/Seq 301.

[0059] Figure 5 shows a block diagram for the transmit path, when data 503 comes from PCI bus 105 and is sent to FB 308. The CRC process/modes, described above are applicable in the transmit path as well.

[0060] The foregoing adaptive aspects of the present invention are implemented using plural registers in RDMA module 306 and TDMA module 305. The register

values are accessible by processors 109 and 112. Figure 7A and 7B show the various register values that are used during a receive (from FB 308) and transmit (from PCI bus 105) operation.

[0061] It is noteworthy that the present invention can be used in compliance with the iSCSI standard and performs the data integrity process steps on iSCSI protocol data units ("PDUs").

[0062] Figure 6 shows a flow diagram of executable process steps for performing data integrity steps, according to one aspect of the present invention.

[0063] In step S601, the process receives data. Data may be received from FB 308 (receive path) or from PCI 105 (transmit path). In step S602, the process determines if CRC is disabled (See Bit 6 in Figures 7A and 7B). If CRC mode is not enabled, the process sends data in step S603.

[0064] If the CRC mode is enabled, then the process selects a particular mode, namely, Append, Validate & Keep, and Validate and remove mode. The modes may be selected by firmware using the plural bits shown in Figures 7A and 7B.

[0065] If the Append mode is selected, then in step S608, CRC is inserted after each data block boundary,

as described above, and then data is sent in step S609 and processed in step S609A.

[0066] If the Validate & Keep mode is selected, then in step S610, CRC is compared, as described above. If an error is found, then an interrupt is sent in step S611 and thereafter, CRC is sent with the data in step S612.

[0067] If the Validate & Remove mode is selected, then in step S613, the process compares the CRC. If an error is detected, then an interrupt is generated in step S614. In step S615 the process removes the CRC after comparison and data is processed in step S616.

[0068] Although the present invention has been described with reference to specific embodiments, these embodiments are illustrative only and not limiting. Many other applications and embodiments of the present invention will be apparent in light of this disclosure and the following claims.